

# RPSGAe - Reduced Pareto Set Genetic Algorithm: Application to Polymer Extrusion

A. Gaspar-Cunha, J.A. Covas

Dept. of Polymer Engineering, University of Minho, Portugal  
gaspar.jcovas@dep.uminho.pt

**Abstract.** In this paper a Multiobjective Optimization Genetic Algorithm, denoted as Reduced Pareto Set Genetic Algorithm with Elitism (RPSGAe), is presented and its performance is assessed. The algorithm is compared with other Evolutionary Multi-Objective Algorithms - EMOAs (SPEA2, PAES and NSGA-II) using problems from the literature and statistical comparison techniques. The results obtained showed that the RPSGAe algorithm has good overall performance. Finally, the RPSGAe algorithm was applied to the optimization of the polymer extrusion process. The aim is to implement an automatic optimization scheme capable of defining the values of important process parameters, such as operating conditions and screw geometry, yielding the best performance in terms of prescribed attributes. The results obtained for specific case studies have physical meaning and correspond to a successful process optimization.

## 1 Introduction

The majority of the real-world optimization problems are multiobjective, since they require the simultaneous satisfaction of several objectives. There are several ways to tackle this. The most usual approach consists of congregating the various individual objectives in a unique function in order to form a single objective optimization problem. In this case, it is necessary to define *a priori* a compromise between the criteria considered. If the relative importance of the criteria is changed a new optimization run needs to be carried out. The other approach takes advantage of the fact that Genetic Algorithms work with a population of points processed in each iteration, yielding a set of non-dominated vectors, denoted as Pareto-optimal solutions. In this case, all the criteria are simultaneously optimized.

After the initial work of Schaffer [1], a considerable number of different Evolutionary Multiobjective Optimization (EMO) algorithms have been proposed. Good reviews have been prepared by Deb [2] and Coello [3]. Generally, these algorithms can be divided in three classes. The first is based on non-Pareto approaches and includes techniques such as aggregating functions [2] and VEGA (Vector Evaluated Genetic Algorithm) [1]. For example a weighted sum (equation 1) can be used:

$$F_i = \sum_{j=1}^q w_j F_{i,j} \quad (1)$$

where  $q$  is the number of criteria,  $F_{i,j}$  is the value of criterion  $j$  for the individual  $i$ ,  $w_i$  is the weight of criterion  $j$  and  $F_i$  is the fitness of the individual  $i$ . Thus, the decision is made before the search, and the solution obtained is a single point. These techniques do not incorporate directly the concept of Pareto optimum, are unable to find some portions of the Pareto frontier and are capable to handle only a small number of criteria. However, they are easy to implement. The second class is based on the concept of Pareto Optimality. It emerged after Goldberg's suggestion that selection should be made using a non-dominated ranking scheme and that diversity should be maintained with the use of a sharing function [4]. For example, the algorithm proposed by Fonseca and Fleming (MOGA) [5] uses a ranking scheme where the rank of each individual corresponds to the number of individuals in the current population by which it is dominated. Fitness sharing is used in order to maintain diversity, together with a mating restriction scheme that avoids crossover between very distant individuals in the search space. Srinivas and Deb [6] implemented a Pareto based ranking scheme in the Non-dominated Sorting Genetic Algorithm (NSGA). They sort the population in various fronts. The non-dominated individuals belonging to the first front are more fitted, hence they are removed from the population and the process is repeated until the entire population is classified. Horn et al. [7] proposed a different algorithm (NPGA) that uses a tournament selection scheme based on the concept of Pareto dominance.

Recently, a third class of EMOAs based on the use of an elite-preserving operator that suppresses the deterioration of the population fitness along the successive generations has been proposed. These algorithms perform sequentially the three basic tasks of fitness assignment, density estimation and archiving. Deb and co-authors [8, 9] suggested an elitist non-dominated sorting GA (known as NSGA-II). The method uses simultaneously an elite preservation strategy and an explicit diversity preserving mechanism. First, an offspring population is created using the parent population, both of size  $N$ . These populations are combined together to form a single population of size  $2N$ . Then, a classification of the population using a non-dominated sorting is performed. Finally, the new population is filled with the individuals of the best fronts, until its size becomes equal to  $N$ . If the population becomes bigger than  $N$ , a niching strategy is used to select the individuals of the last front. The algorithm proposed by Zitzler and Thiele [10], called Strength Pareto EA (SPEA), introduces elitism by maintaining an external population. First, a random initial population of size  $N$  and an empty external population of size  $N_e$  are created. In each generation the solutions belonging to the best front are copied to the external population. Then, the dominated solutions found in this modified population are deleted. When the number of solutions in the external population exceeds  $N_e$ , a clustering algorithm is used to eliminate the more crowded solutions. This algorithm was modified recently, in order to incorporate a fine-grained fitness assignment strategy, a density estimation technique and an enhanced archive truncation method - the SPEA2 algorithm [11]. Corne, Knowles and Oates [12] proposed PESA (Pareto Envelope-based Selection Algorithm), which uses a small internal population

and a larger external population. Initially, an internal population and an empty external population are created. Then, the non-dominated points of the internal population are incorporated in the external population. When a stop criterion is reached, the result will be the non-dominated individuals of the external population. Otherwise, the individuals of the internal population are deleted and new ones are created by crossover and mutation, using as parents the individuals of the external population. Finally, Knowles and Corne [13] introduced an algorithm based on the use of an (1+1) evolution strategy and of an external archive of all the non-dominated solutions. Diversity is maintained through the use of an adaptive grid technique, which is based on a new crowding procedure where the objective space is divided recursively. According to the authors this technique has lower computational cost and the setting of the niche-size parameter it is carried out in an adaptive mode [13].

In this paper, the Reduced Pareto Set Algorithm with Elitism (RPSGAe) is proposed in section 2. Fitness assignment and density estimation are accomplished using a single operator, a ranking function obtained with a clustering algorithm [14]. This algorithm aims also to overcome some limitations of its predecessor, the RPSGA [14, 15], such as the fitness deterioration along the generations and the significant number of parameters, by incorporating elitism. The RPSGAe parameters are assessed in section 3. In section 4 the performance of the RPSGAe is evaluated by comparing its results with those of other algorithms for selected test problems. Finally, the proposed algorithm is used for the automatic optimization of polymer extrusion (section 5).

## 2 Reduced Pareto Set Genetic Algorithm with Elitism

Generally EMOAs replace the selection phase of a traditional Evolutionary Algorithm by a routine able to deal with multiple objectives. In this section, the general structure of EMOAs and the algorithm proposed as selection step for multiobjective optimization will be presented and discussed in detail.

### 2.1 The Structure of EMOAs

The role of optimization is to find the best set of parameters that minimizes or maximizes an objective function, by improving the performance in the direction of some optimal point or points. Since most real-world optimization problems involve the simultaneous satisfaction of several objectives, the result of an optimization algorithm is a set of trade-off solutions that takes into account all the objectives considered. For example, and as illustrated in Fig. 1, if we want to minimize the cost or maximize the performance of a specific system, two single optima exist. However, when both criteria are to be optimized simultaneously, the solution is a set of points denoted as Pareto set, expliciting the trade off between the criteria represented. The decision maker can choose the solution that results from a compromise between the relative satisfaction of the individual criteria.

Alternatively, a multiobjective optimization approach can be applied. In this case the individual criteria are optimized in parallel and the decision about their relative importance is made after the search. The result is the Pareto frontier. The main difference between a Conventional EA and a EMOA is the way fitness is assigned to the individuals of the population. In the former, fitness is directly proportional to the evaluation of the individual solutions. In the EMOA fitness must also take into account the proximity. Therefore, an EMOA must accomplish 3 basic tasks (see also Fig. 2):

1. **Fitness assignment**, to guide the population in the direction of the Pareto frontier, by using a robust and efficient multiobjective selection method;
2. **Density estimation**, to maintain the solutions dispersed along the Pareto frontier, using an operator able to take into account the proximity of the solutions;
3. **Archiving**, to prevent the deterioration of the fitness during the successive generations, by maintaining an external population where the best solutions found so far are retained and are periodically incorporated in the main population.

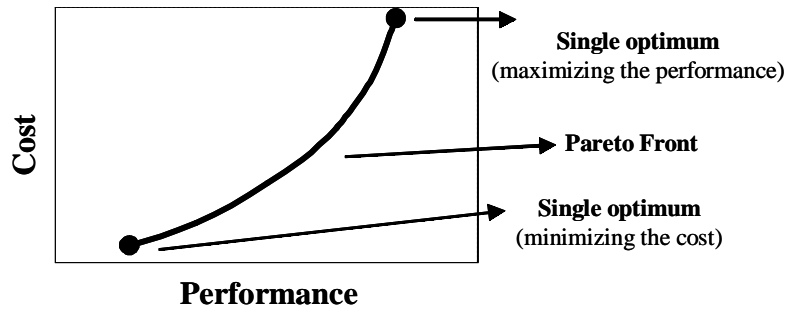
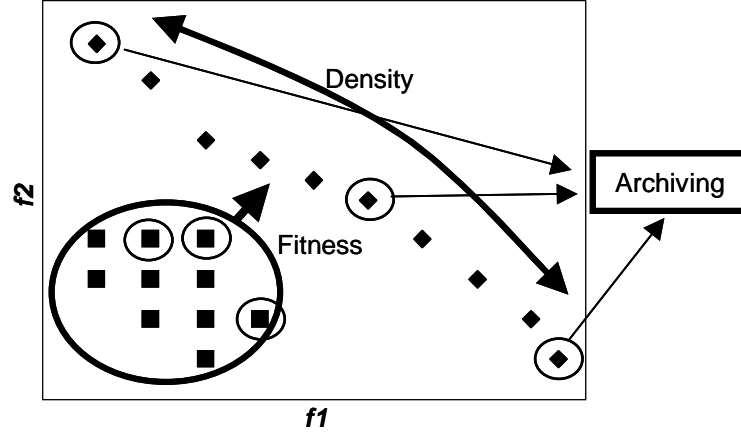


Fig. 1. Minimizing the cost and maximizing the performance of a specific system

## 2.2 Reduced Pareto Set Genetic Algorithm

The Reduced Pareto Set Genetic Algorithm incorporates a technique of clustering the solutions [14], *i.e.*, of reducing the number of solutions on the efficient frontier, while maintaining its characteristics intact. The Complete-linkage method, proposed by Roseman and Gero [16] is used. It consists in comparing the proximity of solutions on the hyper-space, *i.e.*, the similitude or proximity of two solutions is evaluated through a measure of the distance between them, by aggregation of the various distances in each criteria. The RPSGA is based on the assignment of the fitness through a ranking function, where the rank is obtained using a clustering algorithm. The algorithm flowchart is shown in Fig. 3 and comprises the following basic steps:



**Fig. 2.** Multiobjective operators

1. The number of ranks ( $N_{Ranks}$ ) is defined and  $Rank[i]$ , for all  $i$ , is set to 0;
2. For the first rank, *i.e.* current rank number ( $r$ ) equal to 1, the population is reduced down to  $N_R$  individuals ( $N_R$  is the number of individuals of each rank), using a clustering algorithm;
3. Rank 1 is attributed to these  $N_R$  individuals;
4. The algorithm proceeds with the next rank ( $r$  equal to 2) and ends when the number of pre-defined ranks is reached;
5. The fitness of the individual  $i$  ( $F_i$ ) is calculated using a linear ranking function:

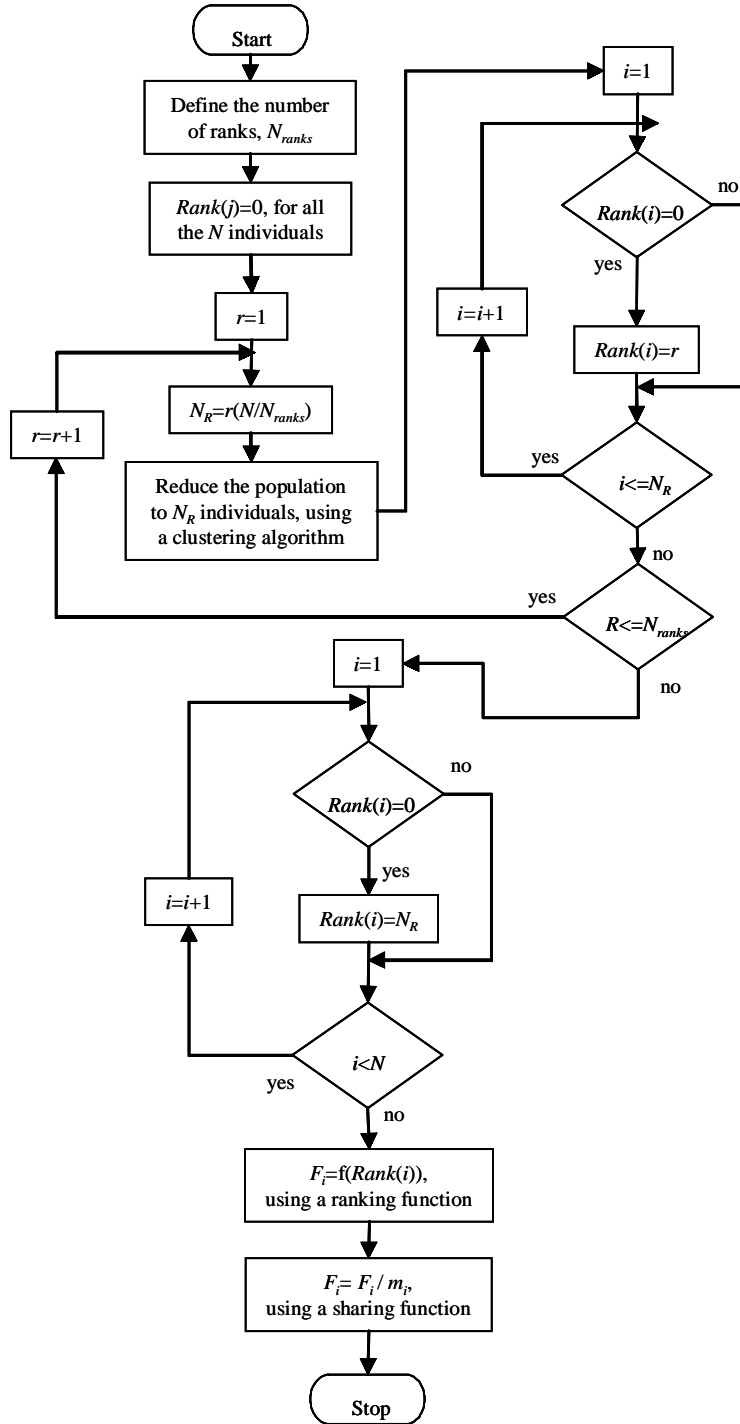
$$F_i = 2 - SP + \frac{2(SP - 1)(N_{ranks} + 1 - Rank[i])}{N_{ranks}} \quad (2)$$

where  $SP$  is the selection pressure ( $1 < SP \leq 2$ ),  $N_{Ranks}$  is the predefined number of ranks and  $Rank[i]$  is the rank attributed to individual  $i$ , together with the application of the sharing function.

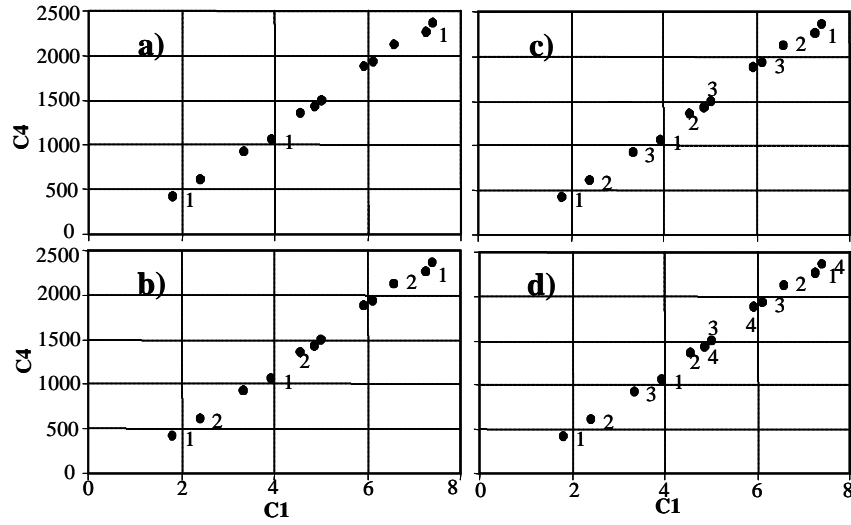
This algorithm takes into account two of the basic EMOA operators, fitness assignment and density estimation. Fitness is calculated using a ranking scheme based on the rank attributed to each individual on the Pareto set. Density is estimated using the sharing function [17]. In the example illustrated in Fig. 4, the aim is to distribute the 12 population individuals by four ranks, using the clustering algorithm, and then to calculate the fitness of all the population individuals.

In appendix is shown the application of the proposed clustering algorithm to reduce the population in order to distribute them by the four ranks previously defined. The following steps must be taken:

- In the first iteration  $r$  is equal to 1 and  $N_R$  equal to 3. Application of the clustering algorithm will reveal that the five individuals identified by open circles in Fig. 4.a will have rank 1.



**Fig. 3.** Reduced Pareto Set Genetic Algorithm (RPSGA) flowchart



**Fig. 4.** Example of application of the RPSGA algorithm

- In the second iteration ( $r$  equal to 2) the value of  $N_R$  is 6. This means that the clustering algorithm reduced the population down to 6 individuals, including the 3 obtained in the first iteration. The three individuals that do not yet have a rank will be labelled as rank 2 (open circles of Fig. 4.b).
- The algorithm proceeds with the two last iterations ( $r$  equal to 3 and 4). In this case, rank 3 and 4 will be attributed to the remaining 6 individuals (Fig. 4.c).
- Fitness is calculated by equation 2, resulting in a fitness of 2 for the individuals with rank 1, 1.5 for the individuals with rank 2, 1 for individuals with rank 3 and 0.5 for the remanning ( $SP=2$ ).
- Finally, and in order to distribute the population uniformly, the sharing function is applied by calculating a new value for the fitness ( $F'_i$ ), taking into account the distance between the individuals [17]

### 2.3 Incorporating Elitism (RPSGAe)

Two modifications to the RPSAGAE algorithm seem important, in order to improve its performance, namely in terms of the fitness deterioration along the successive generations and of the reduction of the number of parameters required. Elitism was introduced by maintaining an external population and prevent fitness deterioration. The number of parameters is reduced by replacing fitness assignment and density estimation by a single operator, a clustering ranking scheme. This approach also avoids the need to pre-define the value of the sharing function, which effectively controls the algorithm performance and needs to be set carefully. The computation sequence of RPSGAe is detailed in Table 1 and illustrated in Figure 5.

The algorithm comprises three main parts. First, the  $N$  population individuals of each generation are reduced to a pre-defined number of ranks ( $r=1, 2, \dots, N_{Ranks}$ ) – steps 1 to 8. Secondly, the value of the objective function is calculated using a ranking function – steps 9 to 12. Finally, an external elitist population is maintained simultaneously, in order to preserve the best solutions found so far and to incorporate periodically some of them in the main population – steps 13 to 16. The steps are detailed in the following paragraph.

**Table 1.** RPSGAe computation sequence

1. Pre-define the number of required ranks,  $N_{Ranks}$ ;
2. Pre-define the size of the Elitist population,  $N_e$ ;
3. Make  $N_{elite}=0$ ;
4. Make  $Rank[i]=0$  for all the  $N$  individuals of the main population;
5. Test of Domination;
6. First iteration,  $r=1$ ;
7. Do:
  - (a) Calculate  $N_R=r*(N/N_{Ranks})$ ;
  - (b) Reduce the population to  $N_R$  individuals using a clustering algorithm;
  - (c) Make  $i=1$ ;
  - (d) Do:
    - If ( $Rank[i]=0$ ) Make  $Rank[i]=r$ ;
    - Make  $i=i+1$ ;
  - (e) While ( $i < N_R$ );
  - (f) Go to the next iteration,  $r=r+1$ ;
8. While ( $r < N_{Ranks}$ );
9. Make  $i=1$ ;
10. Do:
  - (a) If ( $Rank[i]=0$ ) Make  $Rank[i]=N_R$ ;
  - (b) Make  $i=i+1$ ;
11. While ( $i < N$ );
12. Assign a Fitness value for individual  $i$  using an linear ranking function, i.e.,  $F_i = f(Rank[i])$ ;
13. Copy the best  $a*N/N_{Ranks}$  individuals to the elitist population;
14. Make  $N_{elite}=N_{elite} + a*N/N_{Ranks}$ ;
15. If ( $N_{elite} \geq N_e$ )
  - (a) Make  $Rank[i]=0$  for all the  $N_e$  individuals of the elitist population;
  - (b) Apply steps 6. to 12. to the Elitist population;
  - (c) Copy the best  $b*N/N_{Ranks}$  to the main population;
  - (d) Make  $N_{elite}=b*N/N_{Ranks}$ ;
16. Select the individuals for reproduction using roulette-wheel selection;

The algorithm starts with the definition of the number of ranks and of the size of the elitist population and the rank of each individual is initialized to 0 and denoted as  $Rank[i]=0$  (steps 1 to 4). A domination test is carried out in order to define the non-dominated individuals (step 5). The number of individ-



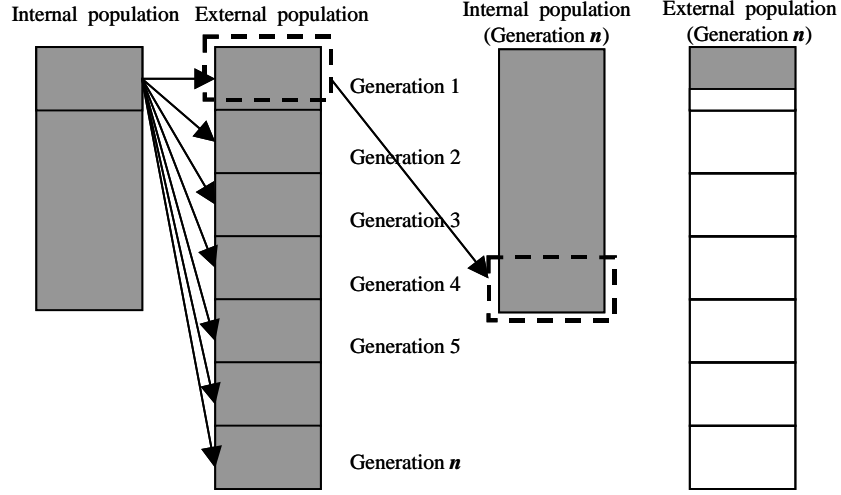
uals is reduced to  $N_R$ , using a clustering algorithm (step 7). This is illustrated with the example shown in appendix. Rank  $r$  (*i.e.*,  $Rank[i]=r$ ) is attributed to those individuals that do not already have a rank (step 7.d), while rank  $N_R$  (*i.e.*,  $Rank[i]=N_R$ ) is attributed to the remaining (step 10.a). At the end the clustering algorithm produce the results of Fig. 4 for the example presented in the appendix. The application of the equation 2 allows the calculation of the global objective function ( $F_i$ ). Elitism is incorporated in the RPSGAe by maintaining an external population of size  $N_e$ . In each generation, the best  $3*N/N_{Ranks}$  individuals (step 13) are copied to the external population, until the number of individuals is lower or equal to  $N_e$  (see Fig. 5). At this point, RPSGAe is used to sort the individuals of the external population (step 15.b). The best  $a*N/N_{Ranks}$  individuals of this population are incorporated in the internal population by replacing those with lower fitness (step 15.c). Simultaneously, only the best  $b*N/N_{Ranks}$  are maintained in the external population (step 15.d). The constants  $a$  and  $b$  are determined empirically in section 3. This algorithm corresponds to the selection phase of a traditional GA, where it can be incorporated.

The order of the algorithm is given by the number of operations required by the domination test and/or by the clustering algorithm. In the first case, each individual needs to be compared with all the others and for all the criteria, which is consistent with an order of  $O(qN^2)$ , where  $q$  is the number of criteria. The clustering algorithm needs  $qN^2$  comparisons and/or calculations for each rank, *i.e.*, it has an order of  $O(N_{Ranks}qN^2)$ . Thus, the global order of the present algorithm is  $O(N_{Ranks}qN^2)$ . The order of the NSGA-II algorithm is equal to the order of the RPSGAe. Since in a "real" world problem, the evaluation of the vectors is generally made resorting to numerical methods, the computation times are mainly associated to this step and, therefore, the order of the optimization becomes meaningless.

### 3 Assessment of the Parameters of the RPSGAe

The correct implementation of the RPSGAe requires the assessment of its main parameters, namely the number of ranks ( $N_{Ranks}$ ), the type of density operator, the limits of indifference of the clustering algorithm, the type of ranking function (linear or exponential), the number of individuals copied in each generation from the internal to the external population, the number of elitist individuals copied to the internal population and the size of the elitist population ( $N_e$ ).

The SPH-2 optimization problem (first equation of Table 3 with  $n=2$ ) will be adopted for this purpose, using 100 parameters and an initial search space ranging from -1000 to 1000. This bi-dimensional minimization problem is characterized by an optimal Pareto frontier with the shape of a quarter circle, centered at [1,1] and with a radius of 1. This problem was chosen due to the difficulty posed to the EMOA in finding the Pareto optimal set, where all the parameters must have the value zero. The comparison of the performance of the various runs for all non-dominated solutions will be made in terms of the average of  $f_1, f_2$  and using the  $S$  metric proposed by Zitzler [19]. All the optimization runs were



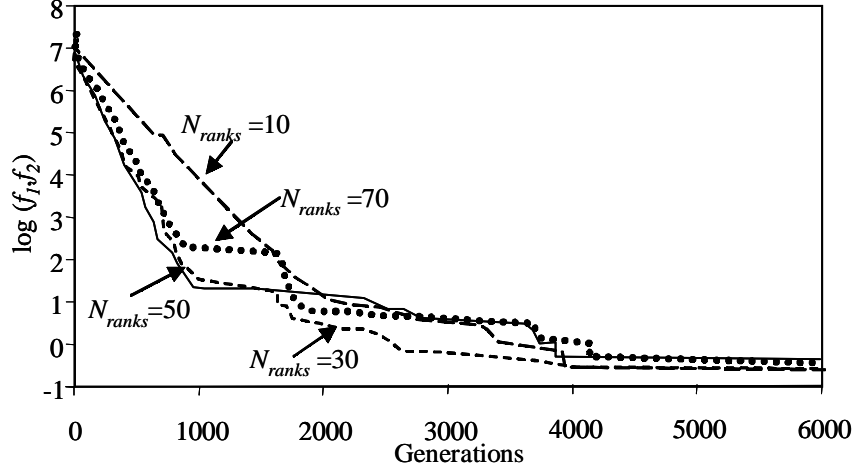
**Fig. 5.** Schematic of the RPSGAe procedure

carried out using a population size of 100, 6000 generations, a roulette wheel selection strategy, a crossover probability of 0.8, a mutation probability of 0.05 and a real encoding to represent the parameters to optimize, together with the SBX operator for recombination and a polynomial distribution for mutation [18].

Fig. 6 presents the evolution of the average of  $f_1, f_2$  as the search proceeds, for various  $N_{ranks}$ . This parameter controls the selection pressure of the RPSGAe algorithm. The higher the  $N_{ranks}$ , the bigger the pressure. The best value for  $N_{ranks}$  is 50 during the initial 1000 generations and 30 for the remaining. Simultaneously, the results along the generations show that, when the objective function value is greater than 20, the number of non-dominated solutions is 1 or 2. This means that the optimization process occurs in two steps. Initially, the algorithm tries to progress towards the Pareto frontier, the solutions are dispersed by the entire search space and only 1 or 2 are non-dominated. Eventually, the algorithm attempts to disperse the solutions in the Pareto frontier, using a confined search space.

Table 2 summarizes the effect of the parameters tested. For that purpose 5 optimization runs were performed for each tested value, while the remaining parameters were kept constant. Thus, a total of 24x5 runs were performed. The effect of each parameter on the improvement of  $f_1, f_2$  was scaled as strong/average/weak. The following can be concluded:

- **Number of ranks ( $N_{ranks}$ )** - As discussed above, when the number of non-dominated vectors is small ( $< 0.1 * N$ )  $N_{ranks}$  must be equal to 50, otherwise the value of 30 can be adopted;
- **Density operator** - Since clustering and sharing yielded similar results, the former will be chosen because of the lower number of calculations necessary;



**Fig. 6.** Influence of the number of ranks ( $N_{ranks}$ ) in the optimization of the SPH-2 problem

- **Limits** - The results showed that the best performance is obtained for values greater or equal to 0.2;
- **Type of ranking function** - The best performance is obtained when the linear ranking function is used;
- **Number of individuals copied from the internal to the elite population** - The best value is  $3*N/N_{Ranks}$ ;
- **Number of individuals copied from the elite to the internal population** - The best result is obtained with  $2*N/N_{Ranks}$ ;
- **Size of the external population** - Identical results were obtained for 100 and 200. The smaller value is chosen, given the smaller number of evaluations of the objective functions.

## 4 Comparative Study

Comparing results produced by different EMO algorithms can be a difficult task, since it is necessary to compare groups of vectors forming a non-dominated set. The metrics used must measure the distance between the resulting non-dominated set and the Pareto-optimal frontier, as well as the uniform distribution of the solutions and the extent of the non-dominated set [20]. Unfortunately, such a metric does not exist [2]. In the present work, a statistical comparison analysis suggested by Fonseca and Fleming [5] and modified by Knowles and Corne [13] was adopted. When two algorithms are compared, two values are generated for each algorithm. The first represents the percentage of the Pareto frontier in which the algorithm is not beaten by the others, while the second

**Table 2.** Influence of the RPSGAe parameters

Parameter	Tested values	Effect
Number of ranks	10; 20; 30; 50; 70	Strong
Density operator	Clustering; Sharing	Weak
Limits	0.01; 0.1; 0.2; 0.3; 0.4; 0.5	Weak
Type of ranking function	linear; exponential	Weak
$N_{internal} \rightarrow$ External pop. <sup>(1)</sup>	$3*N/N_{Ranks}$ ; $2*N/N_{Ranks}$ ; $N/N_{Ranks}$	Weak
$N_{external} \rightarrow$ Internal pop. <sup>(2)</sup>	$3*N/N_{Ranks}$ ; $2*N/N_{Ranks}$ ; $N/N_{Ranks}$	Weak
$N_e$	100; 200; 300	Average

<sup>(1)</sup> No. of individuals copied from the internal to the elite population

<sup>(2)</sup> No. of individuals copied from the elite to the internal population

refers to the percentage of the Pareto frontier on which that specific algorithm beats all the others.

Table 3 presents the test problems used to compare the behavior of the RPSGAe with the PESA[12], NSGA-II[8, 9] and SPEA2[11] algorithms. These problems were chosen not only due to the difficulty that EMOAs have in finding the Pareto-optimal frontier, but also due to the fact that each problem has its characteristics objective functions and Pareto-optimal frontier. The SPH- $n$  problem represents a symmetric unimodal function where the isosurfaces are hyperspheres. Problems with two (SPH-2) and three (SPH-3) objective functions were used. The ZDT6 problem is also unimodal and has a non-uniformly distributed objective space. The QV problem has two main difficulties. One is the multi-modality of the two functions and the other is the significant concavity of the optimal front, associated to a reduction of the density of solutions in the extreme points. The conditions of the optimization runs performed with the RPSGAe algorithm are identical to those used by Zitzler *et. al* [11, 21], *i.e.*, internal and external populations with size 100 and 500,000 evaluations of the objective functions. The results yielded by each of these algorithms, after 30 runs using different seed values for each algorithm and test problem, was also attained from Zitzler *et. al* [21].

The comparison data is summarized in table 4. The RPSGAe is not beaten and beats all the others in 100% of the space for the SPH-2 and SPH-3 problems. In the case of the ZDT6 problem, the RPSGAe algorithm is not beaten in 56.6% of the space and beats all the others in 56.5% of the space. The results are different for the QV problem, where the RPSGAe is not beaten in 11.9% of the space but does not beat the others in any portion of the space. Here, SPEA-2 performs considerably better. The good performance obtained by the RPSGAe for the SPH-2, SPH-3 and ZDT6 problems seems to be due to the fact that during the initial generations, the objective function value is far from the optimal and the number of non-dominated vectors is low (1 or 2, see Fig. 6). In this situation, the RPSGAe is able to impose a high selection pressure, yielding a good evolution of the objective functions. The worst performance obtained for the QV problem

**Table 3.** Test problems ( $m=100$ )

Name	Domain	Objective functions	Ref.
SPH-n	$[-10^3, 10^3]$	$f_j(x) = \sum_{i=1; i \neq j}^m x_i^2 + (x_j - 1)^2$ $j=1, 2, \dots, n$	[1, 22]
ZDT6	$[0, 1]$	$f_1(x) = 1 - \exp((-4x_1) \sin^6(6\pi x_1))$ $f_2(x) = g(x) * \left(1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right)$ $g(x) = 1 + 9 \left(\frac{\sum_{i=2}^m x_i}{m-1}\right)^{0.25}$	[20]
QV	$[-5, 5]$	$f_1(x) = \left(\frac{1}{m} \sum_{i=1}^m (x_i^2 - 10 \cos(2\pi x_i) + 10)\right)^{1/4}$ $f_2(x) = \left(\sum_{i=1}^m ((x_i - 1.5)^2 - 10 \cos(2\pi(x_i - 1.5)) + 10)\right)^{1/4}$	[23]

is probably associated to the difficulty of the RPSGAe to find extreme objective functions values in order to improve the extension of the non-dominated set. Given its good overall performance for the various type of optimization problems considered, the RPSGAe seems a valid option to use in future applications.

**Table 4.** Comparison between the algorithms

Algorithm	PESA	NSGA-II	SPEA-2	RPSGAe
SPH-2	[0, 0]	[0, 0]	[0, 0]	[100, 100]
SPH-3	[0, 0]	[0, 0]	[0, 0]	[100, 100]
ZDT6	[0, 0]	[43.5, 0.1]	[43.3, 0]	[56.5, 56.5]
QV	[35.3, 20.8]	[28.9, 7.7]	[61.2, 36.1]	[11.9, 0]

## 5 Polymer Extrusion

### 5.1 The Process

Extrusion is a major plastics processing technology, yielding wide-spread products such as pipes and profiles, film, sheet, filaments and fibers, electrical wires and cables. Plastics compounding, involving the incorporation of additives in a polymer matrix, or the manufacture of innovative polymer blends and composites, is also carried out in extruders. A conventional plasticating extrusion unit uses an Archimedes-type screw that rotates inside a heated barrel (see Fig. 7 bottom).

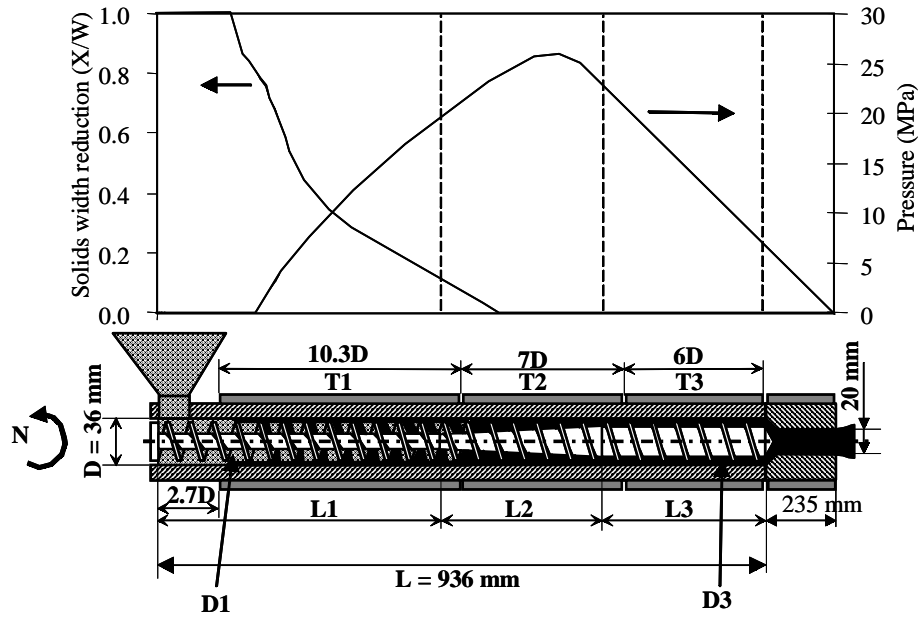


Fig. 7. Extruder geometry (bottom) and solids bed and pressure axial profiles (top)

The solid polymer (in pellets or powder form) flows by gravity in the hopper and progresses along the screw's helical channel due to wall friction forces. Heat conducted and dissipated will induce melting near to the inner barrel wall. The molten material develops an helical recirculating path and accumulates in a pool, segregated from the surviving solids. This melt is homogenized, pressurized and forced to pass through the die, where it is shaped, before being quenched. The global modelling of this process is achieved by linking sequentially the individual stages, using the appropriate boundary conditions. Each zone is described by the relevant governing equations (mass conservation, momentum and energy) together with a constitutive equation describing the melt rheological behavior. A two-dimensional non-Newtonian non-isothermal flow was considered for the melt stages. The calculations are performed in small increments along the screw channel and die, a detailed description being available elsewhere [15, 24–26]. Fig. 7 top illustrates the typical evolution of melting and of pressure generation along the screw axis.

Fig. 8 shows the typical response of an extruder to changes in the operating conditions, or in the geometry. Measures of process performance include mass output, mechanical power consumption, length of screw required for melting, melt temperature, degree of mixing (WATS) and viscous dissipation (quantified by the ratio maximum temperature/barrel temperature). Fig. 8 top evidences that an increase in screw speed from 20 to 50 rpm produces an increase in the values of all the criteria. However, output and power consumption are interre-

lated and depend on the screw rotation frequency. At the higher screw speed range  $WATS$  decreases with increasing screw speed, because there is less channel available for melt distributive mixing. The remaining parameters (length for melting, melt temperature and viscous dissipation) result from the relative contribution of the heat conducted from the heaters and that generated due to the combination of high velocities and viscosity levels. Fig. 8 bottom presents the variation of the same criteria, but now the operating conditions are fixed, the channel depth of the last section having been changed. An increase in the cross-section of the screw channel favours output, but the corresponding decrease in residence time compromises the melting and mixing efficiencies.

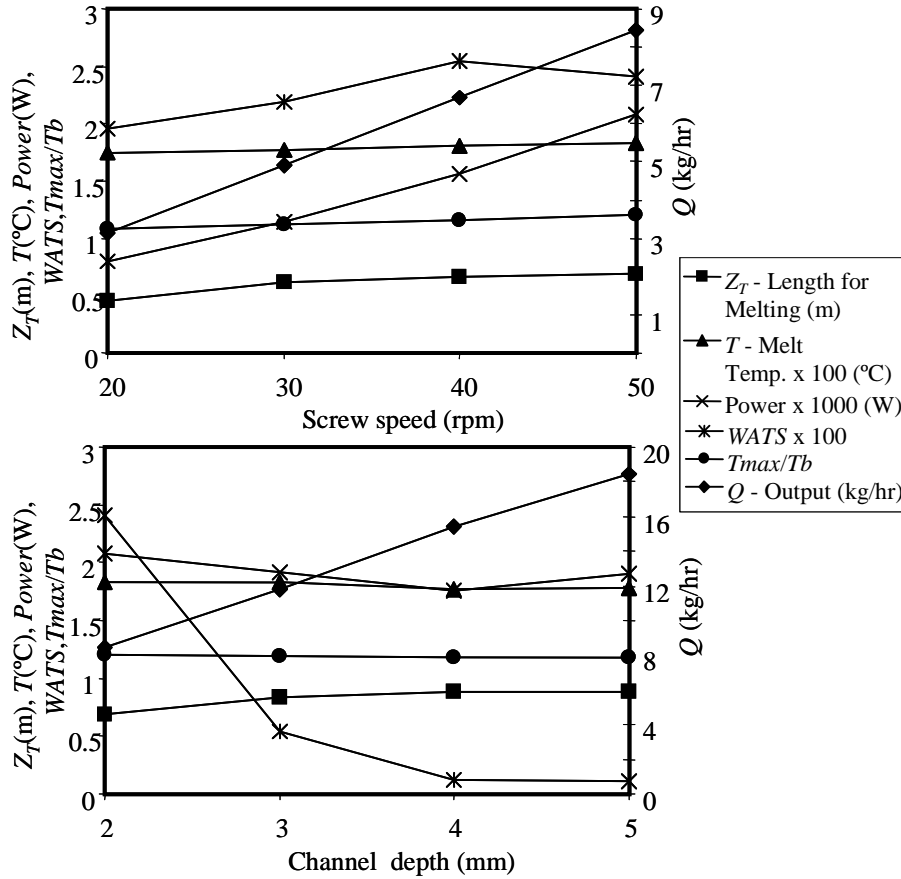


Fig. 8. Effect of screw speed and channel depth on the extruder performance

## 5.2 Multiobjective Polymer Extrusion Problem

The above six criteria will be considered in this section. We wish to select the operating and geometrical variables that maximize mass output and degree of mixing and minimize the remaining (in order to save energy, increase melting efficiency and avoid polymer degradation, respectively), as presented in Table 5. The prescribed range of variation for each criterion ( $X_{min}$  and  $X_{max}$ ) is also presented in Table 5. These limits are due to practical reasons, such as the effective power available, or the fact that no solids must be present in the extrudate (since they compromise the performance of the profile under service conditions). Maximization of mass output implies higher power consumption. Thus, *a priori*, these two objectives are in conflict. In our approach, each point in the search space is a vector  $x$ , representing the four input variables.

**Table 5.** Criteria and corresponding range of variation

Criterion	Aim	$X_{min}$	$X_{max}$
C1- Output (kg/hr)	Maximize	1	10
C2 - Length of screw required for melting (m)	Minimize	0.2	0.9
C3 - Melt temperature ( $^{\circ}\text{C}$ )	Minimize	150	210
C4 - Power consumption (W)	Minimize	0	9200
C5 - WATS	Maximize	0	1300
C6 - Viscous dissipation - $T_{max}/T_b$	Minimize	0	3

### a) Optimization of Operating Conditions

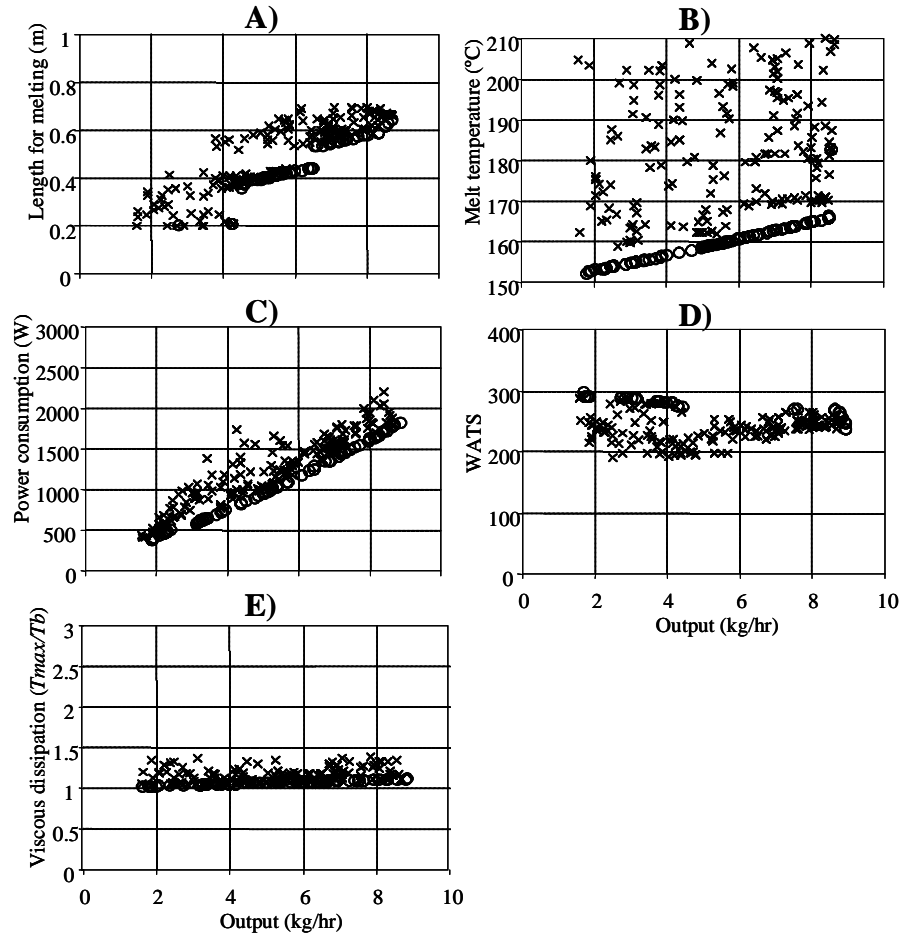
The aim is to optimize the screw speed ( $N$ ), ranging in the interval [10, 50] rpm, and the barrel temperature profile ( $T_1$ ,  $T_2$  and  $T_3$ ), ranging in the interval [150, 210] $^{\circ}\text{C}$  (see Fig. 8). The properties of the polymer and the geometry of the extruder/die system are obviously known. The following GA parameters were used: number of generations, 50, crossover rate, 0.8, mutation rate, 0.05, size of the internal and external populations, 100, limits of the clustering algorithm, 0.2 and  $N_{Ranks}$ , 30. As shown in Table 6, six optimization runs were carried out, only one run per condition was performed, given the computation time required circa 8 hours in a computer with an AMD processor at 2000MHz. Five runs dealing with only two criteria will characterize the behavior of the system, since it is easier to visualize the results in a bi-dimensional space. Mass output will be used in these runs, since it can be considered as the most important criterion. Finally, run 6 considers the simultaneous optimization of all the criteria.

The results are presented in Figures 9 to 12 (the crosses and the open circles represent the initial and non-dominated 50th population, respectively). Fig. 9 illustrates the trade-off between each criteria of runs 1 to 5 against output (criterion C1). The interrelationship between the real extruder parameters to optimize and again output is shown for the case of run 1. All the criteria are



**Table 6.** Optimization runs

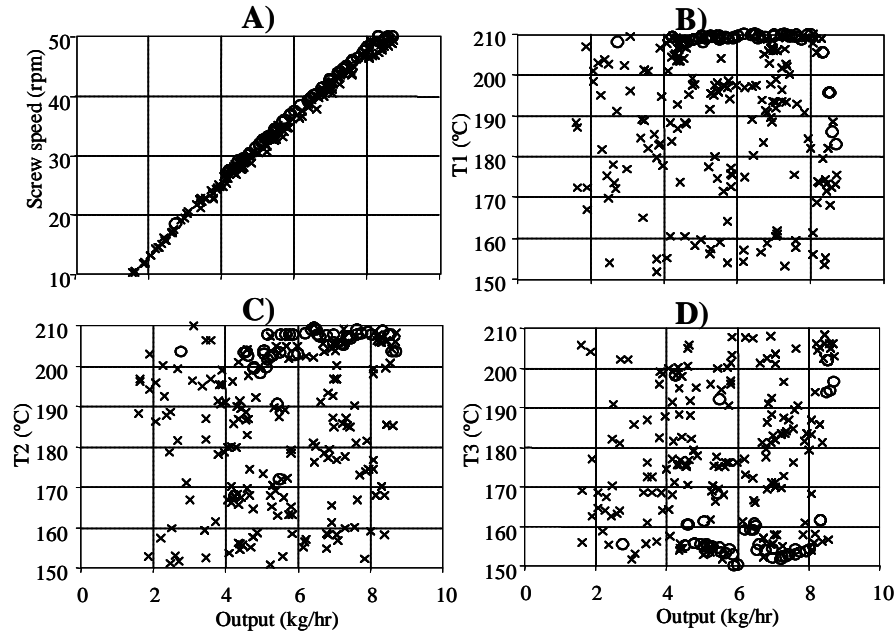
Runs	Criteria used
1	C1 and C2
2	C1 and C3
3	C1 and C4
4	C1 and C5
5	C1 and C6
6	C1 to C6



**Fig. 9.** Optimization of the operating conditions (runs 1 to 5 in Table 6: A) run1; B) run2; C) run3; D) run4; E) run5)

considered in (Fig. 11), which can be directly compared with Fig. 9. Finally, Fig. 12 shows the same data as Fig. 11, but now in the real extruder parameters to optimize domain.

As expected, output conflicts with length required for melting (Fig. 9-A), melt temperature at die exit (Fig. 9-B) and mechanical power consumption (Fig. 9-C). Polymer melting results from the combined effect of heat conduction from the heated barrel and dissipation from friction between the solid polymer and the channel walls. The former is proportional to the residence time, while the latter depends on the screw rotation frequency.



**Fig. 10.** Results for run 1 in the parameters to optimize domain (Table 6)

Fig. 10 presents the results for run 1 in the parameters to optimize domain (*i.e.*,  $N$ ,  $T_1$ ,  $T_2$  and  $T_3$ ). Similar data was also obtained for the other runs. Since output increases with screw speed (Fig. 10-A), residence time decreases. Thus, Fig. 9-A demonstrates that heat conduction prevails in relation to friction effects. Observation of Figures 9-B) and 10 shows that minimization of melt temperature at die exit can be achieved by decreasing barrel set temperatures. Also, lower screw speeds reduce viscous heat dissipation. Increasing screw speed requires more mechanical power (Fig. 9-C). However, lower power consumption at high speeds can be achieved at high barrel temperatures due to the corresponding lower polymer viscosity. In the case of Fig. 9-D, referring to the quality of mixing ( $Q$  *vs.*  $WATS$ ), the Pareto-optimal frontier is discontinuous and  $WATS$  decreases

slightly with an increase in output. Mixing depends on the length of the melt conveying zone and on screw speed. Therefore, the best mixing conditions occur for low and high values of output, i.e., when the length of the melting zone and screw speed are significant. Finally, the results obtained for run 5 ( $Q$  vs. viscous dissipation), presented in Fig. 9-E, are similar to those for run 2. The highest viscous dissipation occurs for high outputs. The comparison made, in all the optimization runs, between the initial population and the 50th generation shows that the optimization algorithm was able to make a good approximation to the Pareto front.

Run 6 considers the simultaneous optimization of the six criteria. In this case, a six-dimensional Pareto frontier in the criteria domain is obtained (Fig. 11). Distinction between dominated and non-dominated solutions is difficult, since points that seem to be dominated in one Pareto frontier are certainly non-dominated in another. However, the use of the statistical comparison technique proposed by Knowles and Corne [13] showed that there are important improvements when the initial population and the final population are compared.

Moreover, it is also difficult to choose a solution from this multidimensional Pareto frontier. Since the location of one possible solution in the various graphical representations is not evident. However, one must remember that a table identifying all the Pareto solutions is available. The decision maker must define a working point (or region) in the criteria domain curves and select the corresponding solution in the parameters to optimize domain represented in Fig. 12. For example, if output is again considered as the most important criterion, point **P1** in Fig. 11 (*i.e.*, the point with the greater output) can be defined and the corresponding solution chosen in Fig. 12. In this case, the extruder must operate with a screw speed of 48.5rpm and a barrel temperature profile of 161/199/195°C. This is done using the tabular form of the solutions represented in Fig. 12. It is clear, however, that this solution is unsatisfactory when criteria such as length for melting, power consumption and viscous dissipation are considered. Alternatively, if length for melting is considered as the most important criteria, point **P2** in Fig. 11 (*i.e.*, the point with the lower length for melting) can be chosen, and the corresponding solution (Fig. 12) selected (screw speed of 27.8rpm and barrel temperature profile of 187/204/199°C). Therefore, graphical and tabular Pareto frontiers are a powerful tool enabling the decision maker to select different solutions representing different compromises between the criteria considered.

### b) Screw Design

In the example to be studied here the aim is to define (see Fig. 8) the length of zones 1 and 2 ( $L_1$  and  $L_2$ , ranging in the intervals [150,400]mm and [250,400]mm, respectively), the internal screw diameter of section 1 and 3 ( $D_1$  and  $D_3$ , ranging in the intervals [20,26]mm and [26,32]mm, respectively), the screw pitch ( $P$ , ranging in the interval [30,42]mm) and the flight thickness ( $e$ , ranging in the interval [3,4]mm) that will optimize the criteria already identified in Table 6 for each run. The operating conditions are fixed at  $N=50$ rpm and  $T_i=170^\circ\text{C}$ . The

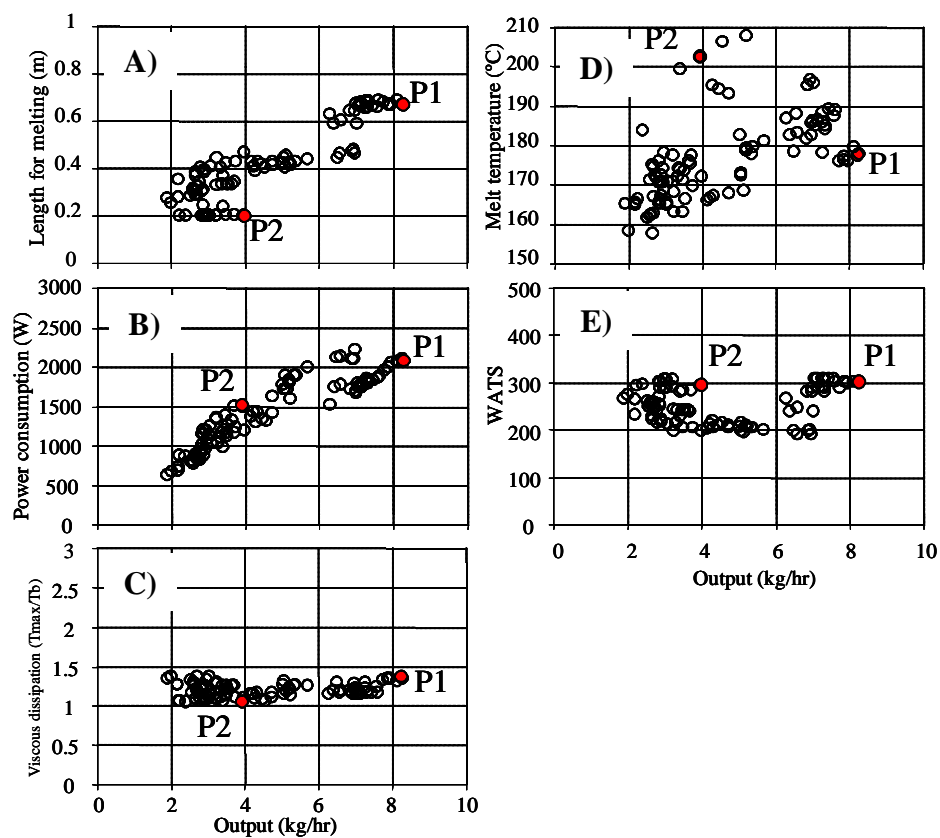


Fig. 11. Results for run 6 in the criteria domain (Table 6)

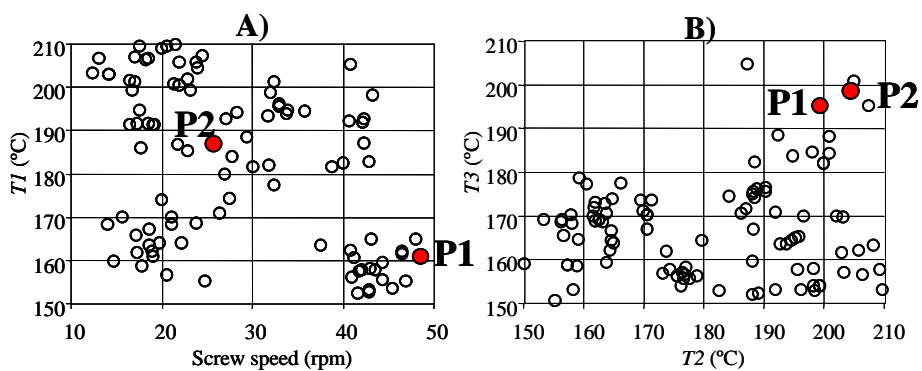


Fig. 12. Results for run 6 in the parameters to optimize domain (Table 6)

constraints referred above for the criteria (Table 5) remain valid. The results are shown in Figures 13 to 14. Fig. 13 refers to run 1 (Table 6) in the criteria

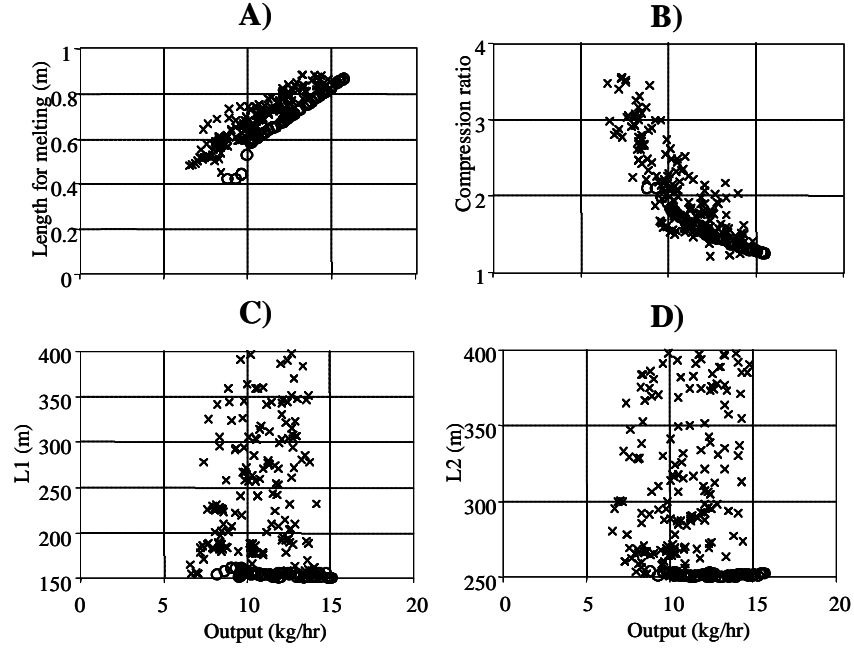


Fig. 13. Results for screw design (run 1 of Table 6)

domain (A) and in the parameters to optimize domain (B to D), respectively. Given the referred conflicting character between output and length for melting high outputs are obtained with screws with small compression ratios (Fig. 13.b), *i.e.*, for screws with small variations in the internal diameter. Simultaneously,  $L_1$  and  $L_2$  should be short (Fig. 13 C and D) to promote early melting. The results for runs 2 to 5, are presented in Fig. 14, and can be explained with the reasoning used above.

Fig. 15 shows the screw geometries obtained for run 6 when output and *WATS*, respectively, are considered as the most important of the six criteria. A screw with small compression ratio and high channel depth maximizes output (Fig. 15 top), as it offers less resistance to polymer melt flow. Conversely, a screw with high compression ratio and a shallow channel maximizes the degree of mixing (Fig. 15 bottom), given the high velocities in the relatively long pumping section. Obviously, these two criteria are conflicting, as the very different screw geometries suggest.

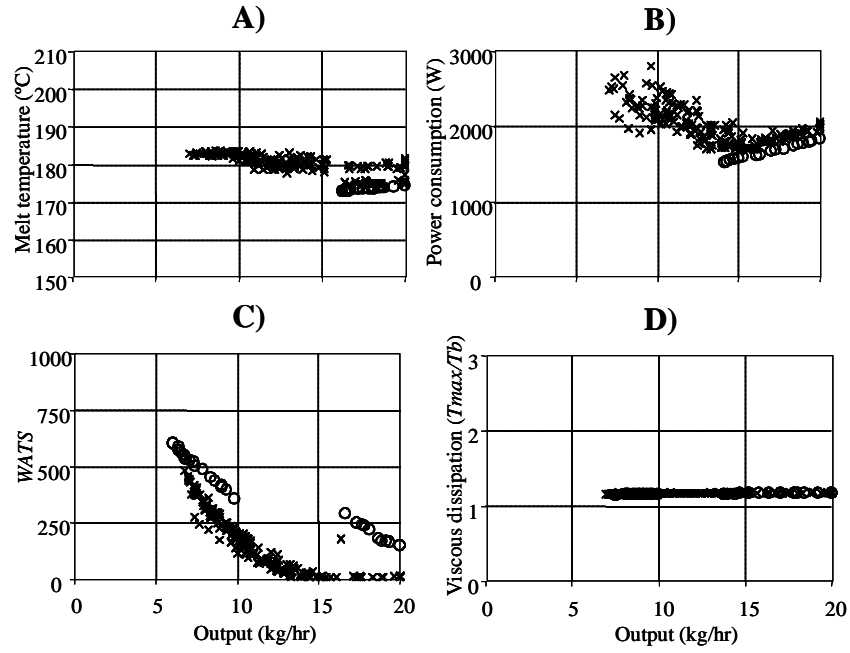


Fig. 14. Results for screw design (runs 2 to 6 of Table 6)

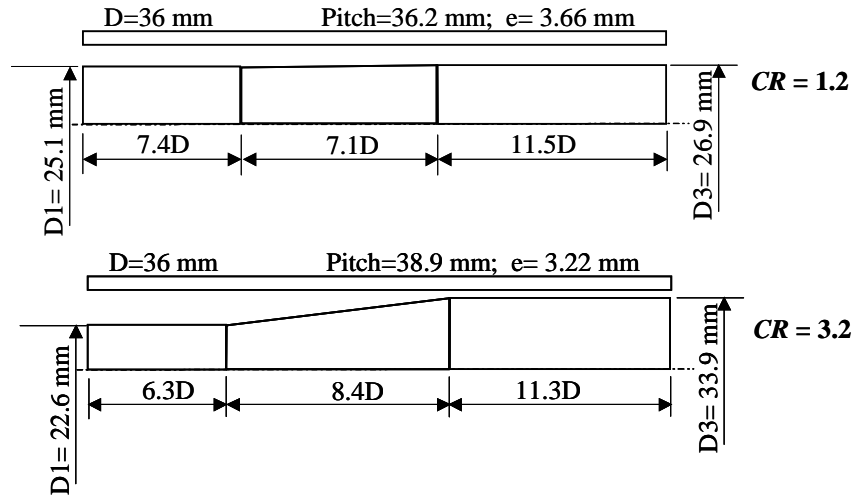


Fig. 15. Screw geometries for maximization of output (top) and WATS (bottom) – run 6, of Table 6

## 6 Conclusions

In this work, an elitist multiobjective genetic algorithm, denoted as Reduced Pareto Set Genetic Algorithm with Elitism (RPSGAe), has been proposed and assessed. The main characteristic of this algorithm is the use of a clustering technique to rank the individuals as a function of their fitness and, simultaneously, to distribute them uniformly along the Pareto frontier. A study performed using efficient statistical comparison techniques showed that, for the problems considered, the RPSGAe algorithm is generally able to find solutions with a higher level of performance.

The RPSGAe was applied to the optimization of the operating conditions and to the design of screws for polymer extrusion. The results obtained showed that the optimization methodology proposed is able to find solutions with physical meaning.

The RPSGAe can be easily applied to problems containing a significant number of criteria, where the number of dominated vectors in the population becomes very small or nil. Since RPSGAe is independent of the domination test, its application is not constrained to problems that depend on the number of dominated vectors to progress towards the Pareto frontier.

### Acknowledgements

This work was supported by the Portuguese Fundação para a Ciência e Tecnologia under grant POCTI/34569/CTM/2000.

## References

1. Schafer, J.D.: Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms, Ph. D. Thesis, Nashville, TN, Vanderbilt University (1984)
2. Deb, K.: Multi-Objective Optimisation using Evolutionary Algorithms, Wiley (2001)
3. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer (2002)
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley (1989)
5. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, Proc. Fifth Int. Conf. on Genetic Algorithms, Morgan Kaufman (1993) 416–423
6. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation **2** (1995) 221–248
7. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proc. First IEEE Conf. on Evolutionary Computation (1994) 82–87
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-Objective Optimisation: NSGAI, Proceedings of the Parallel Problem Solving from Nature VI (PPSNVI) (2000) 849–858
9. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGAI, IEEE Transactions on Evolutionary Computation **6** (2002) 182–197

10. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation* **3** (1999) 257–271
11. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, TIK report no. 103, Swiss Federal Institute of Technology, Zurich, Switzerland (2001)
12. Corne, D.W., Knowles, J.D., Oates, M.J.: The Pareto-Envelope based Selection Algorithm for Multiobjective Optimisation, *Proceedings of the Parallel Problem Solving from Nature VI (PPSNVI)* (2000) 869–878
13. Knowles, J.D., Corne, D.W.: Approximating the Non-dominated Front using the Pareto Archived Evolutionary Strategy, *Evolutionary Computation Journal* **8** (2000) 149–172
14. Gaspar-Cunha, A., Oliveira, P., Covas, J.A.: Use of Genetic Algorithms in Multi-criteria Optimization to Solve Industrial Problems, *Seventh Int. Conf. on Genetic Algorithms*, Michigan, USA (1997)
15. Gaspar-Cunha, A.: Modelling and Optimisation of Single Screw Extrusion, Ph. D. Thesis, University of Minho, Guimares, Portugal (2000)
16. Roseman, M.A., Gero, J.S.: Reducing the Pareto Optimal Set in Multicriteria Optimization, *Eng. Optim.*, **8** (1985) 189–206
17. Goldberg, D.E., Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Optimization, *Proc. Second Int. Conf. on Genetic Algorithms* (1987) 41–49
18. Deb, K., Agrawal, R.B.: Simulated Binary Crossover for Continuous Search Space, *Complex Syst.*, **9** (1995) 115–148
19. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, PhD Thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (1999)
20. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, **8** (2000) 173–195
21. Zitzler, E. (2002). Test Problems for Multiobjective Optimizers. Retrieved January 01, 2002, from <http://www.tik.ee.ethz.ch/zitzler/testdata.html>
22. Laumann, M., Rudolph, G., Schwefe, H.-P.: Mutation Control and Convergence in Evolutionary Multi-Objective Optimization, *Proceedings of the 7th International Mendel Conference on Soft Computing* (2001)
23. Quagliarella, D., Vicini, A.: Coupling Genetic Algorithms and Gradient Based Optimization Techniques, D. Quagliarella, J. Priaux, C. Poloni, G. Winter (Eds.), *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, Wiley, Chichester (1997) 289–309
24. Gaspar-Cunha, A.; Covas, J.A.; Oliveira, P.: Optimization of Polymer Extrusion With Genetic Algorithms, *IMA J. of Math. Appl. in Business and Industry*, **9** (1998) 267–277
25. Covas, J.A.; Gaspar-Cunha, A.; Oliveira, P.: An Optimization Approach to Practical Problems in Plasticating Single Screw Extrusion, *Polym. Eng. and Sci.*, **39** (1999) 443–456
26. Gaspar-Cunha, A., Covas, J.A.: The Design of Extrusion Screws: An Optimisation Approach, *Intern. Polym. Process.*, **16** (2001) 229–240



## Appendix

In this appendix, the application of the RPSGA algorithm (described in detail in Fig. 3) to the case study of Fig. 4 is illustrated. For this purpose, it is necessary to detail the step identified as "Reduce the population down to  $N_R$  individuals, using a clustering algorithm" (see Fig. 3), which corresponds to reducing the Pareto set by the method proposed by Roseman and Gero [16].

This method known, as Complete-linkage [16], clusters similar vectors in all or in some criteria. It involves the sequence illustrated in Fig. A-I, which can be sub-divided in two stages, organizing the vectors and performing the reduction, respectively. Initially, each population individual,  $N$ , is ascribed to a separate group,  $p$ . For each criterion,  $j$ , an indifference limit,  $L_j$ , is defined. The vectors are sorted by ascending order and the difference between the values of the criteria of the last and the first groups,  $R_j = C_{j,p} - C_{j,1}$ , is calculated. Then, for each group,  $k$ , the difference between adjacent groups (or vectors),  $D_{j,k}$ , is determined. In the second stage merging of similar and/or elimination of dominated vectors is carried out. For this purpose, the set of the vector(s) with the lowest value of  $D_{j,k}$  (if lower than  $L_j$ ) is defined. If this set is not empty, merging adjacent vectors is carried out, recalculation of the value of  $D_{j,k}$  and elimination of any dominated vector ( $p=p-1$ ) are performed. These calculations are repeated until  $p$  is lower or equal to the number of required vectors,  $N_R$ .

Table A-I contains the data of Fig. 4, but considering three criteria ( $q=3$ ), output (C1), melt temperature (C3) and power consumption (C4) – see also Table 5. The population contains 12 individuals ( $N=12$ ), and it will be divided into 4 ranks ( $N_{Ranks}=4$ ) – first step of the RPSGA algorithm (Fig. 3). Initially the rank of each population individual is set to 0 (see Fig. 3). Then, for each rank,  $r=1, 2, \dots, N_{Ranks}$ ,  $N_R$  is calculated (for  $r=1$   $N_R$  is equal to 3). The next step consists in the application of the clustering algorithm (Fig. A-I) to reduce the population from 12 to 3 individuals (Fig. 3). Table A-II presents the sorting of the vectors, the definition of the groups and the calculation of  $D_{j,k}$  and  $R_j$ , for each criterion  $j=1, 2, q$  (according to the first stage Fig. A-I. The indifference limits are set to a high value ( $L_j=0.2$ ), in order that the stop criterion of the clustering algorithm is  $N_R=3$ . Then, the second stage of the algorithm Fig. A-I is applied. The minimum difference is 0.5, for melt temperature ( $j=2$ ) and for group 6 ( $k=6$ ), which defines the first set A ( $D_{2,6}$ ).  $D_{2,6}$  is lower than  $L_j$  (0.2). Therefore, the groups 6 and 7 (for this criterion) are merged into group 6 and the differences recalculated.

The next step of the Complete-linkage method is the domination test. Both vectors are non-dominated, as can be seen in Table A-III, since for criterion C1, H (6) dominates J, (9) but for criterion C3 J, (4) dominates H, (7). Thus, no vector is eliminated. The algorithm carries out 9 iterations, at which  $\min D_{j,k} = 5.7$  (see Table A-IV, where the vectors eliminated were shadowed). The second stage of the algorithm continues until  $N_R=3$ , and the computing sequence of Fig. 3 is recovered.

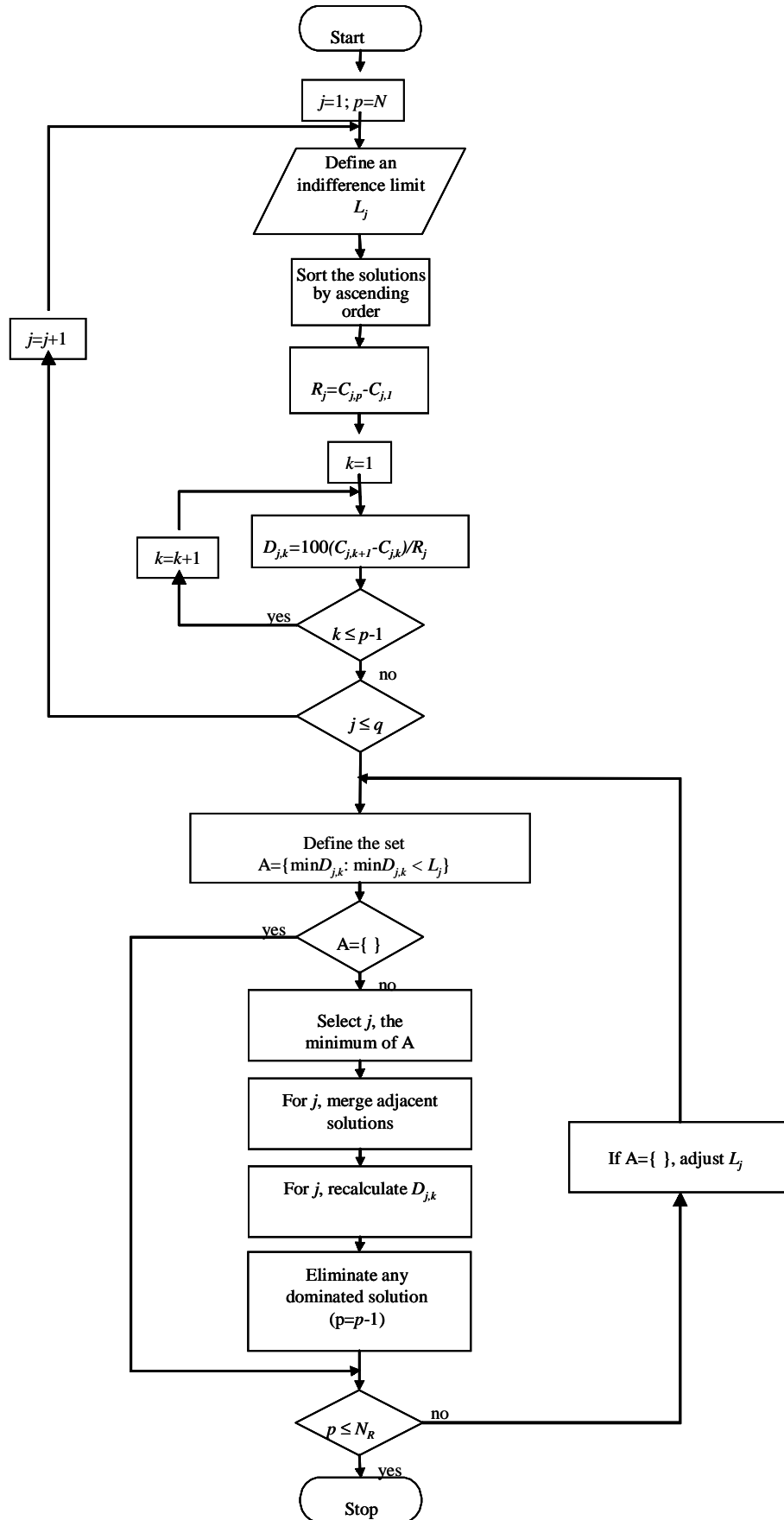


Fig. A-I. Complete-linkage method, developed by Roseman and Gero [16]

Since in this example the stop criterion is the number of vectors the algorithm must continue until the number of vectors becomes equal to 3. Table A-IV presents the set for iteration 9, where the eliminated solutions are shadowed. The algorithm of Fig. A-I ends and the control of the program returns to the algorithm of Fig. 3. Rank 1 (*i.e.*,  $Rank(j)=r$ ) is attributed to the individuals that are not eliminated. The RPSGA algorithm proceeds with the next rank ( $r=2$ ) and  $N_R=6$ . Table A-V shows the final results.

**Table A-I.** Pareto optimal set for extrusion example

Individual	C1- Output (kg/hr)	C3- Melt Temperature (°C)	C4- Power Consumption (W)
A	5.92	185	1875
B	4.57	180	1359
C	2.41	174	615
D	3.95	186	1058
E	3.34	177	921
F	1.81	175	413
G	7.42	201	2361
H	4.88	183	1425
I	5.03	182	1501
J	6.12	183	1935
K	6.58	184	2118
L	7.27	202	2251

**Table A-II.** First iteration

Group	Individual	C1	$D_{jk}$ (%)	Group	Individual	C3	$D_{jk}$ (%)	Group	Individual	C4	$D_{jk}$ (%)
1	F	1.81	10.8	1	L	202.1	4.0	1	G	2361	5.7
2	C	2.41	16.5	2	G	201.0	52.9	2	L	2251	6.8
3	E	3.34	10.8	3	D	186.3	6.0	3	K	2118	9.4
4	D	3.95	11.1	4	A	184.7	4.1	4	J	1935	3.0
5	B	4.57	5.6	5	K	183.5	1.1	5	A	1875	19.2
6	H	4.88	2.5	<b>6</b>	<b>H</b>	<b>183.2</b>	<b>0.5</b>	6	I	1501	3.9
7	I	5.03	16.0	7	J	183.1	4.1	7	H	1425	3.4
8	A	5.92	3.4	8	I	182.0	6.8	8	B	1359	15.5
9	J	6.12	8.3	9	B	180.1	11.3	9	D	1058	7.0
10	K	6.58	12.3	10	E	176.9	6.2	10	E	921	15.7
11	L	7.27	2.5	11	F	175.2	3.1	11	C	615	10.4
12	G	7.42		12	C	174.4		12	F	413	
$R_j$		<b>5.61</b>				<b>27.78</b>				<b>1947</b>	

**Table A-III.** Test for domination

Condition	C1	C3	C4
H	6	-	7
J	9	-	4

**Table A-IV.** Iteration 9

Group	Individual	C1	$D_{jk}$ (%)	Group	Individual	C3	$D_{jk}$ (%)	Group	Individual	C4	$D_{jk}$ (%)
1	F	1.81	11.1	1	L	202.1	56.9	1	G	2361	
2	C	2.41	16.9	2	G	201.0		2	L	2251	7.2
3	E	3.34	11.1	3	D	186.3	10.1	3	K	2118	10.0
4	D	3.95	11.4	4	A	184.7		4	J	1935	23.6
5	B	4.57	8.3	5	K	183.5	5.7	4	A	1875	
5	H	4.88		5	H	183.2		5	I	1501	7.7
6	I	5.03	19.9	5	J	183.1		6	H	1425	
7	A	5.92		6	I	182.0	6.8	6	B	1359	16.4
7	J	6.12	8.5	7	B	180.1	11.3	7	D	1058	7.5
8	K	6.58	12.7	8	E	176.9	9.3	8	E	921	16.7
9	L	7.27		9	F	175.2		9	C	615	11.0
9	G	7.42		9	C	174.4		10	F	413	

**Table A-V.** Final results

Individual	A	B	C	D	E	F	G	H	I	J	K	L
Rank	4	2	2	1	3	1	4	4	3	3	2	1
Fitness	0.5	1.5	1.5	2.0	1.0	2.0	0.5	0.5	1.0	1.0	1.5	2.0